

I claim:

1. A method of simulating system conditions at a kernel-level, comprising:

5 intercepting an operating system call from an application at a kernel-level;

 determining whether the operating system call was called from a process that was identified for failure emulation;

10 if the operating system call was called from a process that was identified for failure emulation, consulting user loaded rules and returning results to the operating system call according to the user loaded rules; and

 if the operating system call was not called from a
15 process that was identified for failure emulation, calling a native operating system service routine associated with the operating system call.

2. The method of claim 1, wherein the determining
20 includes at least determining whether an identifier of the process was communicated previously as being a process for failure emulation.

3. The method of claim 1, wherein the user loaded
25 rules include type of failure to emulate, frequency of failure, or error codes to return, or combination thereof.

4. The method of claim 1, wherein the intercepting is transparent to the process for failure emulation.

30

5. A method of simulating system conditions at a kernel-level, comprising:

identifying one or more processes to a kernel-level module for which to emulate failures;

transmitting one or more failure rules to the kernel-level module, the one or more failure rules associated with

5 the one or more processes;

activating the kernel-level module; and

running the one or more processes.

6. The method of claim 5, wherein the identifying
10 includes at least transmitting one or more process identifiers to the kernel-level module, the one or more process identifiers associated with respective one or more processes for which failure emulation is to be performed.

15 7. The method of claim 5, further including deactivating the kernel-level module.

8. A program storage device readable by machine, tangibly embodying a program of instructions executable by
20 the machine to perform method of simulating system conditions at a kernel-level, comprising:

intercepting an operating system call from an application at a kernel-level;

determining whether the operating system call was
25 called from a process that was identified for failure emulation;

if the operating system call was called from a process that was identified for failure emulation, consulting user loaded rules and returning results to the operating system
30 call according to the user loaded rules; and

if the operating system call was not called from a process that was identified for failure emulation, calling

a native operating system service routine associated with the operating system call.

9. The program storage device of claim 8, wherein the
5 determining includes at least determining whether an identifier of the process was communicated previously as being a process for failure emulation.

10. The program storage device of claim 8, wherein
10 the user loaded rules include type of failure to emulate, frequency of failure, or error codes to return, or combination thereof.

11. The program storage device of claim 8, wherein
15 the intercepting is transparent to the process for failure emulation.

12. A system for simulating system conditions at a kernel-level, comprising:

20 a user-space module operable to transmit one or more process identifiers and one or more rules associated with the process identifiers for emulating failure conditions at a kernel-level; and

a kernel-level module operable to intercept system
25 call, and further operable to determine whether the system call was invoked from one or more processes identified by the one or more process identifiers and if the system call was invoked from the one or more processes identified by the one or more process identifiers, the kernel-level
30 module further operable to generate a return result according to the one or more rules, and if the system call was not invoked from the one or more processes identified

by the one or more process identifiers, the kernel-level module further operable to call native operating system service routine associated with the system call.

- 5 13. The system of claim 12, wherein the user-space module further includes an application programming interface that communicates with the kernel-level module.

10